MPC823 Design Checklist

Version 1.3+

When integrating the MPC8xx family of microprocessors into an application, the following items can help guide you through the process. These guidelines originated from common problems that were found while debugging and operating actual MPC8xx-based systems.

Before Getting Started

- **Find the Mask Number and Version of Your Chip**

  Each chip has different sets of numbers etched onto it and one of these sets is the mask and revision number for that particular chip. The mask number and the revision number are combined into one. For example, 0F98S. 0 is the revision number and F98S is the mask number. You need this information in order to find the errata for that version of the chip. Obtaining the appropriate errata will help you design your product more efficiently. Once you know your mask and revision numbers, see the MPC823 Device Errata for your particular chip errata.

  In addition, the feature differences between early revs of the MPC823 are significant. This is another reason to find the mask number and version of your chip.

System Considerations

- **Always Provide a Development Interface Port on Your Design**

  The MPC823 Family Application Development System (MPC823FADS) has a 10-pin header that you can use to connect to your target board. With this header installed, you can perform initial testing and follow-up debugging. By connecting a FADS board to your PC, you can use our PC hosted debugger (MPC8Bug Debugger Package - MPC8BUGDBG) to manipulate registers on your target board, just as you would directly on the FADS board. However, you must remove the MPC823 device from its socket on the FADS board. To minimize noise and maintain system integrity, use the shortest 10-wire cable between the FADS board and your target board. Other third-party tools use this 10-pin header for their debugger capability as well.

  The DSDI and DSCK pins of the 10-pin header are configured during reset to select certain modes of the development port system interface. If you want to pull these pins low, then you should use 1Kohm pull-downs so that low logic levels are recognized on all versions of the MPC823.

  The MPC823FADS boards can be an invaluable tool for developing software or you can simply use the accompanying schematics as a template. Our web site contains the documentation and schematics for the MPC823FADS boards, which can be ordered from your local Motorola sales office.

- **IRQ0~ is really NMI~**

  It is mentioned in the manual that IRQ0~ causes a non-maskable interrupt, i.e., reset. Do not connect anything to this pin that you do not want causing a system reset.

- **Memory upgrades for FADS/Compatible Systems**

  Any 16MB, 32-bit 72-pin 60ns EDO DRAM SIMM should work with our ADS or FADS systems. We recommend ordering from Southland Micro Systems (www.southlandmicro.com). The part number for 16MB SIMMs is **SGE 4X32T6E**.

  Note that 5V DRAMs are needed since the FADS boards are designed for it. If you choose 3.3V DRAM, solder a zero-ohm resistor or a short on R83.

Memory Considerations

See *Sections 15a, 15b, 15c, and 15d of Memory Controller* of the MPC823 User's Manual for additional information.

- **Increase Your Chip-Select Drive Capability**

  You can get more drive capability on the CS2 and CS3 signals by having each of them output on two pins. CS2 can be driven on the normal CS2 pin and GPL_A2 and CS3 can be driven on the normal CS3 pin and GPL_A3. The CSx pins can also be used as RAS lines where drive capability is critical in gaining the last nanosecond from the memory system. Program the SIUMCR register (located in *Section 12 System Interface Unit* of the MPC823 User's Manual) to implement this double drive function.

- **Do Not Rely Solely on the R/W Pin**

  The timing of the R/W pin may not be the best choice for the memory type you have chosen. Remember that the R/W timing is fixed and the WEx signals are usually a better choice. If the timing of the WEx signals is not suitable, you can use the more flexible user-programmable machine of the memory controller to get precise timing on these signals.

Clocking Considerations

Using the following suggestions can improve the stability of your clock and minimize skew. See *Section 5 Clocks and Power Control* of the MPC823 User's Manual for more detailed information.

- **Choose Oscillators Instead of Crystals**

  If you can afford the power dissipation and higher cost of using an oscillator instead of a crystal for the clock input source, choose an oscillator. Often, crystal circuits must be tweaked after the die goes through a geometry shrink. If you need to use both an oscillator and a crystal on the same design and if the crystal is the main source for the system clock, you should gate off the external clock. You can use oscillators and crystals simultaneously on the MPC823. However, we recommend against using the crystal on the EXTAL and XTAL pins and, to prevent excessive jitter, you should leave the oscillator free running into the EXTCLK pin. If you are not using a crystal, then ground the EXTAL pin. If the crystal is only used for the real-time clock and the main clock source is from the oscillator, then gating is not required.

- **Place the Crystal Within 0.5 Inches of the MPC823**

  The crystal and the XFC capacitor must be as close to the chip as possible. See *Table 5-4* of the MPC823 User's Manual for detailed information.

- **Do Not Use an RC Filter on the VDDSYN Pin**

  You should use an LC filter instead. See *Figure 5-2* of the MPC823 User's Manual for an details.

Electrical Considerations

- **Determine the proper value for your External Filter Capacitor (XFC)**

  The following table illustrates how to compute the minimum, maximum, and recommended values of the XFC capacitor, as a function of the MF field in the PLPRCR register. Please consult *Section 5 Clocks and Power Control* of the MPC823 User's Manual for more information. **Note: This table is valid only for MPC823 CDR2 version silicon (i.e., any Mask Number not of form: xF98S).**

MF Range	Recommended Capacitance	Minimum Capacitance	Maximum Capacitance	UNIT
MF <= 4	680 * (MF+1) - 120	580 * (MF+1) - 100	780 * (MF+1) - 140	pF
MF > 4	1100 * (MF+1)	830 * (MF+1)	1470 * (MF+1)	pF

- **Remember Not All MPC823 Pins Are 5 Volt Friendly**
- **Do Not Use a 2.2 Volt Source for Frequencies Greater Than 25MHz**

  If 2.2 is the voltage source for the VDDL pins, then the maximum frequency of the device is 25MHz. Otherwise, the VDDL power plane should be connected to a 3.3V source.

- **Use Multiple Power and Ground Planes**

  You should use at least one ground plane, one 3.3V VCC plane, and one 5V VCC plane (if 5V parts exist in the system). In addition, at least four signal planes would normally be used to ensure easy routing.

- **Decouple the Analog Power Plane**

  The VDDSYN pin is routed to a via that punches down to a mini plane (a peninsula on a signal layer), which is decoupled from VSSSYN with 10µF and 0.1µF parallel caps and also attaches to the 3.3V plane through a 8.2mH inductor. The 3.3V and 5V planes have a keep-out region in the PLL power plane area. You should separate the PLL VSSSYN from global ground by splitting the plane with a connection point as close to the power supply as possible. On other layers, signals are kept out of this area. You should also place the PLL loop filter and decoupling caps close to the MPC823.

- **Improve DRAM Signal Integrity**

If you are using DRAM SIMMs in your design, use series dampening resistors on all memory control lines to improve signal integrity.

- **Use Pull-Ups and Pull-Downs**

  We suggest following pins must be pulled-up/pulled-down with an appropriate value to ensure higher frequency operation:

Pin	Resistor Value
BR*	4.7K
BB*	2K
BI*	4.7K
TA*	2K
TEA*	2K
DSCK	1K Pull-Down

- **Do Not Leave Unused Input Pins Floating**

  Floating input and tri-state input pins may cause excessive current draw in low-power modes. The address and data bus signals are OK because the MPC8xx can pull them down internalls with the FIOPD bit in the PLPRCR. All other unused input pins and tri-state input pins need to pulled high, ie TSIZx, R_W, DPx, IRQ0-7. Also, coming out of reset the GPL_A4~/UPWAITA/AS~ and GPL_B4/UPWAITB are inputs and, if unused, need to be set as outputs or pulled high. Floating pins can cause over 100µA of current each.

  Due to varying voltage levels on floating inputs, not every device will indicate a higher than expected current consumption, so it is important that you ensure that your design is properly terminated. You may want to test a large sample (>30) for verification of your low-power configuration and design.

- **When You Are Not Using the Port Pins, Configure Them as Outputs**

  The parallel port pins do not have internal pull-ups or pull-downs. Therefore, unused parallel I/O pins should be configured as outputs after reset and left unconnected.

- **Avoid contentions on lines**

  Do not set a pin high while it is being externally driven low.

- **TMS and TDI**

  TMS and TDI need to be either disconnected or pulled high. They have weak internal pull-ups.

- **Consider MODCKx Pin Configuration**

  You can configure the MODCKx pins to become outputs after reset by driving them with three-state buffers enabled by HRESET, but not PORESET since it has a slow rising edge.

- **Connect TRST~ to PORESET~**

  Connect TRST~ to PORESET~ through a diode, pointing towards PORESET~.

- **Driving SRESET***

  The MPC823 can drive the soft reset pin (SRESET*) as an output. It is driven by the MPC823 any time that PORESET* or HRESET* is asserted, and also if a RESET command is delivered to the debug port.

- **Properly configure CPM General Purpose IO (GPIO) pins**

  When CPM inputs are externally driven by a device that enters its own low-power mode with tri-stated pins, change the GPIO pin to outputs, and set the value so that it will not cause a problem.

  - Signal the external device to shutdown.
  - Change the GPIO direction to output.
  - Enter a low power mode.
  - On wakeup from low power mode, change GPIO direction

    Wakeup external device.
- **Decide Whether Or Not You Need PCMCIA Buffers**

Although there are buffers in PCMCIA design examples these buffers on the address or data bus are only required if the address or data bus is heavily loaded, if you are designing a 2-slot PCMCIA system, or if the board needs to be protected by a PCMCIA card hot plug-in. Keep in mind that the MPC823 supports *one* PCMCIA slot.

- **Optimize Memory-to-Memory IDMA Transfers**

  If you need to use the IDMA in memory-to-memory mode, you must assert the DREQx pin. You can either ground the pin or use a parallel I/O pin to selectively enable the DREQx function. Using the parallel I/O pin is recommended, which saves the RISC microcontroller overhead since it has to poll the IDMA buffer descriptors to see if the valid (V) bit has been set.

>Reset Considerations

See *Section 4 Reset* of the MPC823 User's Manual for additional information.

- **Actively Drive the Data Bus During Reset Configuration**

  Driving the data bus with an active device instead of pull-ups during reset configuration ensures that the appropriate value is written into the configuration register. If you do or do not use pull-ups on the data bus and default hard reset configuration word is acceptable, tie the RSTCONF* pin high. If you want to change the hard reset configuration word to something other than the default, drive the appropriate data bus lines low with an active device and connect the RSTCONF* pin to ground. If the default hard reset configuration word is not acceptable and power consumption is a concern, drive the appropriate data lines high with an active device and connect the RSTCONF* pin to ground.

Software Considerations

- **Remember PowerPC Byte Lane Architecture**

  If you are using an 8-bit wide memory/EPROM, attach the device to byte lane zero (D0-D7). In PowerPC architecture, D0-D7 is the highest order byte lane on the data bus and D0 is the highest order pin of that byte lane. D0-D7 corresponds to write enable 0 (WE0) and Byte Select A or B 0 (BS_x0). D0-D7 is associated with WE0 and BS_x0. D15-D8 is associated with WE1 and BS_x1, D23-D16 is associated with WE2 and BS_x2, and D31-D24 is associated with WE3 and BS_x3. The most significant bit is D0 and the least significant bit is D31.

- **Remember the PowerPC Addressing Convention**

  On the MPC823, the highest order address pin is A6, and the lowest order address pin is A31.

- **Using MMU lite with a display**

  With a few simple additions, MMU lite (MPC821MMULDRV) can almost remove the need for using dynamic translations. More than one MPC8XX TLB can point to the same physical memory, so long as their virtual memory addresses are different. For example, we could setup TLB 1 to make all of DRAM with the caches on and set to copyback. TLB 2 can be set with caches on and writethrough, and TLB 3 with cache inhibit. Each one will need its own virtual address, but they can all point to the same physical DRAM. For example:

TLB	Virtual Address	Real Address	Caching Mode
1	0x0000_0000	0x0000_0000	Copyback
2	0x1000_0000	0x0000_0000	Writethrough
3	0x2000_0000	0x0000_0000	Inhibit

  Normal variables can be accesses through TLB1, the display through TLB2, and CPM buffers through TLB3. Remember the CPM always deals with real addresses.

- **Don't Cache Internal Memory space**

  Internal Memory Space, configured in the Hard Reset Configuration Word, must not be in a data-cached region. If it is, erratic operation may occur. See *Section 4 Reset* in the MPC823 User's Manual.

- **When using 2 MB FLASH and larger, set ~IIP=1**

  Setting the ~IIP=1 in the Hard Reset Configuration Word will cause the Initial PowerPC Interrupt Table base address to be 0x0. This will

ensure that the Interrupt Table, at boot time, will always be in the same location in the FLASH, namely the beginning, regardless of the size of the FLASH. See *Section 4 Reset* in the MPC823 User's Manual.

- **External Interrupt Setup and Access**

  For external interrupt pins, if a request signal is a pulse, the interrupt request pin should be configured to "edge detect mode". See the subsection on Interrupt Configuration in *Section 12 SIU.* This ensures that the interrupt will be recognized even if interrupts are temporarily blocked or disabled by the software. The interrupt service routine (ISR) should clear the edge status flag after the ISR is entered and prior to setting the EE bin in the MSR (Machine Status Register). If the ISR waits until after the EE bit is set, a second interrupt may be taken.

  If a request signal is a "standard handshake", the assertion is asynchronous, but the negation occurs upon request from the ISR. This ensures that the interrupt is taken and the source of the interrupt is known. The timing with respect to the EE bit is the same.

  To avoid spurious interrupts, interrupt masks should not be set while interrupts might be sent to the core. Likewise, no interrupts should be disabled while the interrupt might be pending at the core. That way, when the core responds to the interrupt request, the request will still be pending and the core can determine the source of the interrupt. To accomplish all of the above, the EE bit should be disabled when masks are set or when interrupt enables are cleared.

- **Writing to the ICTRL register**

  If you use the mtspr instruction to set the Ignore First Match (IFM) bit of the ICTRL register at the same time you set an instruction breakpoint on this instruction, the chip will behave unpredictably. The workaround is to disable instruction breakpoints when setting the IFM bit.

- **Overlapping Parameter RAM Tables for Ethernet and SPI (MPC823 Only)**

  The address range in Parameter RAM (PRAM) for Ethernet is (DPRAM_Base + 0x1d00 ... 0x1da3). The address range in Parameter RAM for SPI is (DPRAM_Base + 0x1d80 ... 0x1daf). If you require concurrent operation of SPI and Ethernet then download the Microcode Patch which relocates the SPI PRAM.

- **Don't let Internal Memory Space and the PowerPC Interrupt Table overlap**

  Make sure the ISB (Initial Internal Space Base) and the PowerPC Interrupt Table base location do not overlap each other. The IIP (Initial Interrupt Prefix) bit in the Hard Reset Configuration Word determines the initial PowerPC Interrupt Table base address, after reset. The Interrupt Table includes the Reset Vector (offset 0x100), where execution begins after reset. The ISB field in the Hard Reset Configuration Word determines the base address of the Internal Memory Space. If there is an overlap, the PowerPC core may fetch instructions from Internal Memory space, leading to erratic operation. See *Section 4 Reset* in the MPC823 User's Manual. An example of an illegal combination of IIP and ISB in the Hard Reset Configuration Word would be:

IIP	ISB
0x1	0x00

Ethernet Considerations

- **The MC68160 Is Not Recommended for New Designs**

  The Motorola MC68160 Enhanced Ethernet Transceiver, which is featured in the MPC823FADS, is no longer recommended for new designs. Level One and AMD are alternative sources of ethernet transceivers for use with the MPC823.

LCD Considerations

- **Choose an LCD panel that works with a 4 clock cycle HSYNC pulse**

  The MPC823 LCD controller is specifically designed to operate with panels that take an HSYNC pulse of 4 clock cycles in length. See the *Section 18 LCD* in the MPC823 User's Manual.

- **Caching the LCD/video frame buffer**

  If you are manipulate bit fields on the LCD frame buffer for graphic drawings, it's recommended you mark the frame buffer as cacheable write-through. This will help cut down the bandwidth consumption due to read accesses and yet still keep the picture coherence with what in the cache. Otherwise, you should mark the frame buffer as cache-inhibited.

  An easy way to setup the caches with the LCD controller is covered under Using MMU lite with a display.

- **Differences Between the MPC823 and MPC821 LCD Controllers**

  The MPC823 uses 4 bits to encode each of Red, Green, and Blue (RGB) for **TFT displays only** in the LCD Color RAM words. The MPC821 uses 3 bits to encode each color. This is the only difference.

- **Understand the Factors that Limit the Maximum Display Resolution**
  1. **HPC** -- Horizontal Pixel Count (this is a field in the LCHCR), which is related to the number of pixels per line on the panel, by Table 18-2 in the MPC823 User's Manual. The maximum value for HPC is 2047.
  2. **VPC** -- Vertical Pixel Count (this is a field in the LCVCR), which is related to the number of lines on the panel by Table 18-3 in the MPC823 User's Manual. The maximum value for VPC is 1023.
  3. **BNUM** -- Memory Bursts per Frame (this is a field in the LCCR). The maximum value for BNUM is 32767.

     From the maximum value allowed by BNUM, we can compute maximum values for H (number of lines), and W (number of pixels per line), assuming BPIX = 8.

     BNUM = W * H * BPIX / 128 = 32767

     H * W = 524272 but, W = 4/3 * H assuming a standard 4/3 aspect ratio (e.g., 640x480, 1024x768, etc...)

     4/3 * H * H = 524272

     H = 627 (approx.)
     W = 836 (approx.)

     So, 800x600 is the maximum standard resolution at 8 bits per pixel.

  4. **Bus Bandwidth** -- We recommend not exceeding 45% bus utilization for the LCD panel.

     Assuming a design using SDRAM with a timing of 4-1-1-2 (8 clocks per burst), running at 50 MHz system frequency, and again assuming 8 bits per pixel.

     LCD Panel bandwidth = 45% * 50 MHz / 8 clocks per burst * 16 bytes per burst = 45 MB/s

     Assume a frame rate of 100 frames/s

     Number of bytes per frame = 45 MB/s / 100 frames/s = 450,000

     Number of pixels per frame = 450,000 again, assuming 8 bits per pixel

     H * W = 450000 again, assuming a standard 4/3 aspect ratio, as in the above example,
     4/3 * H * H = 450000

     W = 580 (approx.)
     H = 774 (approx.)

     However, by running slightly above the recommended bus utilization (at 48%), you could again select a resolution of 800x600.